Solving Polynomial Systems

Taylor Brysiewicz (Texas A&M University)

February 16, 2019

GIGEM 2019
Texas A&M University
Polynomial Systems

Polynomial systems show up all of the time

- Algebra (Algebraic geometry)
Polynomial Systems

Polynomial systems show up all of the time

- Algebra (Algebraic geometry)
- Combinatorics (Toric varieties)
Polynomial Systems

Polynomial systems show up all of the time

- Algebra (Algebraic geometry)
- Combinatorics (Toric varieties)
- Cryptography (Elliptic curve cryptography)
Polynomial Systems

Polynomial systems show up all of the time

- Algebra (Algebraic geometry)
- Combinatorics (Toric varieties)
- Cryptography (Elliptic curve cryptography)
- Applications
Polynomial Systems

Polynomial systems show up all of the time

- Algebra (Algebraic geometry)
- Combinatorics (Toric varieties)
- Cryptography (Elliptic curve cryptography)
- Applications
  - Optimization [Ble12]
Polynomial Systems

Polynomial systems show up all of the time

- Algebra (Algebraic geometry)
- Combinatorics (Toric varieties)
- Cryptography (Elliptic curve cryptography)
- Applications
  - Optimization [Ble12]
  - Frame theory [CS]
Polynomial Systems

Polynomial systems show up all of the time

- Algebra (Algebraic geometry)
- Combinatorics (Toric varieties)
- Cryptography (Elliptic curve cryptography)

Applications
- Optimization [Ble12]
- Frame theory [CS]
- Chemical reaction networks [Fei19]
Polynomial Systems

Polynomial systems show up all of the time

- Algebra (Algebraic geometry)
- Combinatorics (Toric varieties)
- Cryptography (Elliptic curve cryptography)
- Applications
  - Optimization [Ble12]
  - Frame theory [CS]
  - Chemical reaction networks [Fei19]
  - Kinematics [HSW17]
Polynomial Systems

Polynomial systems show up all of the time

- Algebra (Algebraic geometry)
- Combinatorics (Toric varieties)
- Cryptography (Elliptic curve cryptography)

Applications

- Optimization [Ble12]
- Frame theory [CS]
- Chemical reaction networks [Fei19]
- Kinematics [HSW17]

Many many other places
Polynomial Systems

Polynomial systems show up all of the time

- Algebra (Algebraic geometry)
- Combinatorics (Toric varieties)
- Cryptography (Elliptic curve cryptography)

Applications
  - Optimization [Ble12]
  - Frame theory [CS]
  - Chemical reaction networks [Fei19]
  - Kinematics [HSW17]

- Many many other places
Polynomials in GIG’EM so far

- Eigenvalues are solutions to polynomial equations (Amudhan)
Polynomials in GIG’EM so far

- Eigenvalues are solutions to polynomial equations (Amudhan)
- Number of solutions to my favorite polynomial system is a multinomial coefficient (Pablo...)

- We saw $q$-polynomials. Honorable mentions: Bessel, Jacobi, Hahn, Laguerre polynomials (Gallen)
- Compact quantum groups like $SU_q(2)$ are real parts of solutions to polynomial equations (John).
- Amoebas come from polynomials (Nida).
- Taylor approximations (are polynomials) used for Ito’s Lemma (Sean).
- You can encode neural networks being convex as vanishing sets being empty over finite fields (Alex).
- Border rank of matrix multiplication tensors come from polynomial equations (Kashif)
Polynomials in GIG’EM so far

- Eigenvalues are solutions to polynomial equations (Amudhan)
- Number of solutions to my favorite polynomial system is a multinomial coefficient (Pablo...)
- We saw $q$-polynomials. Honorable mentions: Bessel, Jacobi, Hahn, Laguerre polynomials (Gallen)
Polynomials in GIG’EM so far

- Eigenvalues are solutions to polynomial equations (Amudhan)
- Number of solutions to my favorite polynomial system is a multinomial coefficient (Pablo...)
- We saw $q$-polynomials. Honorable mentions: Bessel, Jacobi, Hahn, Laguerre polynomials (Gallen)
- Compact quantum groups like $SU_q(2)$ are real parts of solutions to polynomial equations (John).
Polynomials in GIG’EM so far

- Eigenvalues are solutions to polynomial equations (Amudhan)
- Number of solutions to my favorite polynomial system is a multinomial coefficient (Pablo...)
- We saw $q$-polynomials. Honorable mentions: Bessel, Jacobi, Hahn, Laguerre polynomials (Gallen)
- Compact quantum groups like $SU_q(2)$ are real parts of solutions to polynomial equations (John).
- Amoebas come from polynomials (Nida).
Polynomials in GIG’EM so far

- Eigenvalues are solutions to polynomial equations (Amudhan)
- Number of solutions to my favorite polynomial system is a multinomial coefficient (Pablo...)
- We saw $q$-polynomials. Honorable mentions: Bessel, Jacobi, Hahn, Laguerre polynomials (Gallen)
- Compact quantum groups like $SU_q(2)$ are real parts of solutions to polynomial equations (John).
- Amoebas come from polynomials (Nida).
- Taylor approximations (are polynomials) used for Ito’s Lemma (Sean).
Polynomials in GIG’EM so far

- Eigenvalues are solutions to polynomial equations (Amudhan)
- Number of solutions to my favorite polynomial system is a multinomial coefficient (Pablo...)
- We saw $q$-polynomials. Honorable mentions: Bessel, Jacobi, Hahn, Laguerre polynomials (Gallen)
- Compact quantum groups like $SU_q(2)$ are real parts of solutions to polynomial equations (John).
- Amoebas come from polynomials (Nida).
- Taylor approximations (are polynomials) used for Ito’s Lemma (Sean).
- You can encode neural networks being convex as vanishing sets being empty over finite fields (Alex).
Polynomials in GIG’EM so far

- Eigenvalues are solutions to polynomial equations (Amudhan)
- Number of solutions to my favorite polynomial system is a multinomial coefficient (Pablo...)
- We saw $q$-polynomials. Honorable mentions: Bessel, Jacobi, Hahn, Laguerre polynomials (Gallen)
- Compact quantum groups like $SU_q(2)$ are real parts of solutions to polynomial equations (John).
- Amoebas come from polynomials (Nida).
- Taylor approximations (are polynomials) used for Ito’s Lemma (Sean).
- You can encode neural networks being convex as vanishing sets being empty over finite fields (Alex).
- Border rank of matrix multiplication tensors come from polynomial equations (Kashif)
Polynomial Systems

\( \mathbb{K} \): a field
\( n \in \mathbb{N} \): number of variables
\( R = \mathbb{K}[x_1, \ldots, x_n] \): polynomial ring
\( f_1, \ldots, f_m \in R \): some polynomials

**Goal:** Understand the set

\[ V(f_1, \ldots, f_m) = \{(x_1, \ldots, x_n) \in \mathbb{K}^n | f_i(x_1, \ldots, x_n) = 0 \forall i\} \]

**Subgoal:** Understand the case when \( V(f_1, \ldots, f_m) \) consists of finitely many points. (zero dimensional polynomial solving)

(for this talk, we will assume there are finitely many solutions)
Easy case: One polynomial $f(x) = \sum_{i=0}^{d} c_i x^i \in \mathbb{K}[x]$ in one variable.

How many roots are there?:
Easy case: One polynomial \( f(x) = \sum_{i=0}^{d} c_i x^i \in \mathbb{K}[x] \) in one variable.

How many roots are there?:
If \( \mathbb{K} \) is algebraically closed: \( d \) with multiplicity.
Easy case: One polynomial $f(x) = \sum_{i=0}^{d} c_i x^i \in \mathbb{K}[x]$ in one variable.

How many roots are there?:
If $\mathbb{K}$ is algebraically closed: $d$ with multiplicity.
This talk: $\mathbb{K}$ will be $\mathbb{C}$
Easy case: One polynomial $f(x) = \sum_{i=0}^{d} c_i x^i \in K[x]$ in one variable.

How many roots are there?:
If $K$ is algebraically closed: $d$ with multiplicity.
This talk: $K$ will be $\mathbb{C}$

But wait...what does it even mean to solve a system?
Starting easy

Easy case: One polynomial \( f(x) = \sum_{i=0}^{d} c_i x^i \in \mathbb{K}[x] \) in one variable.

How many roots are there?:
If \( \mathbb{K} \) is algebraically closed: \( d \) with multiplicity.
This talk: \( \mathbb{K} \) will be \( \mathbb{C} \)

But wait...what does it even mean to solve a system?
(1) Give algebraic numbers via minimal polynomial
Starting easy

Easy case: One polynomial \( f(x) = \sum_{i=0}^{d} c_i x^i \in \mathbb{K}[x] \) in one variable.

**How many roots are there?**
If \( \mathbb{K} \) is algebraically closed: \( d \) with multiplicity.
This talk: \( \mathbb{K} \) will be \( \mathbb{C} \)

But wait...what does it even mean to solve a system?
(1) Give algebraic numbers via minimal polynomial
(2) Give approximations of numbers as decimals with option to sharpen
Solving symbolically

Input:

\[ f_1, \ldots, f_m \in \mathbb{C}[x_1, \ldots, x_n] \]

1. Set \( I = \langle f_1, \ldots, f_m \rangle \)
2. Compute a Gröbner basis \( g_1, \ldots, g_M \) for \( I \) (with respect to lexicographic monomial ordering)
3. \( g_M \) will be a polynomial in \( x_n \) only. Solve this univariate polynomial to compute all possible last coordinates of a solution.
4. \( g_{M-1} \) will only have \( x_{n-1} \) and \( x_n \), substitute each possible \( x_n \) to produce a univariate polynomial in \( x_{n-1} \)
5. et cetera

Output: A list of equations with the same solutions such that you can back substitute one variable at a time.
Symbolic example

\[ f = x^2 + y^2 - 5, \quad g = xy - 2 \]

Tell Macaulay2 or some other program to compute a Gröbner basis

Gröbner basis:

\[ \{2x + y^3 - 5y, y^4 - 5y^2 + 4\} \]

Possible y coordinates: \{-1, 1, -2, 2\}

Solutions: (1, 2), (1, -2), (-1, 2), (-2, 1)

Software: Macaulay2 [GS02], SAGE [Ste19], Singular, Maxima, Mathematica
Solving numerically

**Idea:** You give me some system to solve. I (for a moment) ignore your request and solve my own. Then I deform my solutions to yours.

**Example:** You give me

\[ f = x^2 + y^2 - 5, \quad g = xy - 2 \]

I solve my favorite instead

\[ \hat{f} = x^2 - 1, \quad \hat{g} = y^2 - 1 \]

Then I deform along a homotopy

\[ F = (t)f + (1 - t)\hat{f}, \quad G = (t)g + (1 - t)\hat{g} \]

This is done by solving an ordinary differential equation using predictor corrector methods.

Software: Bertini [BHSW], PHCpack [Ver], NumericalAlgebraicGeometry.m2 [Ley09], HomotopyContinuation.jl [Bre18]
Such a start system always has $\prod \deg(f_i)$ solutions. But the end system may not. This means some go “off to infinity” during the homotopy.

Remark: There are different start systems depending on the structure of your equations, but many times you will have an overcount of the number of solutions.
Monodromy Algorithm

Computing solutions to a system via monodromy
Monodromy Algorithm

Suppose we know a point on a torus.
Suppose we know a point on a torus.
Monodromy Algorithm

Take a random linear slice through that point.
Monodromy Algorithm

Take a random linear slice through that point.
Monodromy Algorithm

Pick another random linear slice.
Monodromy Algorithm

Pick another random linear slice.
Monodromy Algorithm

Use homotopy continuation to follow your known point to the new slice.
Monodromy Algorithm

Use homotopy continuation to follow your known point to the new slice.
Monodromy Algorithm

Repeat
Monodromy Algorithm

Repeat
Monodromy Algorithm

Repeat
Monodromy Algorithm

Then return to the original slice
Monodromy Algorithm

Look! Another point!
Monodromy

- Monodromy works with nonlinear varieties too
- You don’t overcompute too much (contrary to using homotopy method)
- The software for this method is quite good:
  - MonodromySolve.m2
  - HomotopyContinuation.jl
- Depending on situation, one can boost this algorithm a lot (if you know there is a natural action on your solutions, like complex conjugation)
## Comparisons

<table>
<thead>
<tr>
<th>Method</th>
<th>Comparison table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symbolic</td>
<td>Yes</td>
</tr>
<tr>
<td>Numerical</td>
<td>Not really (but certifiable)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th># variables matters a lot</th>
<th>degrees of polynomials matters</th>
<th>Parallelizable</th>
<th>Works well over C</th>
<th>Easy to use</th>
<th>Largest system I've solved</th>
<th>Can certify real roots</th>
<th>Can certify rational roots</th>
<th>Metaphor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact</td>
<td>Doesn't matter much</td>
<td>Matters</td>
<td>No</td>
<td>Works well over C</td>
<td>Definitely</td>
<td>≈ 5000</td>
<td>Yes</td>
<td>Yes</td>
<td>Download whole book</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Matters</td>
<td></td>
<td></td>
<td></td>
<td>≈ 200000</td>
<td>Yes</td>
<td>Working on it</td>
<td>Read page at a time</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Easy</td>
<td></td>
<td></td>
<td></td>
<td>Finite fields</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Comparisons

<table>
<thead>
<tr>
<th>Method</th>
<th>Comparison table</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Symbolic</td>
<td>Numerical</td>
</tr>
<tr>
<td>Exact</td>
<td>Yes</td>
<td>Not really (but certifiable)</td>
</tr>
<tr>
<td># variables</td>
<td>matters a lot</td>
<td>doesn’t matter much</td>
</tr>
</tbody>
</table>

### Degree of Polynomials
- Matters a bit
- Matters

### Parallelizable
- No
- Yes

### Works well over
- C
- Not really...

### Easy to use
- Definitely
- Takes some engineering

### Largest system I’ve solved
- ≈ 5000
- ≈ 200000

### Can certify real roots
- Yes
- Yes

### Can certify rational roots
- Yes
- Working on it

### Metaphor
- Download whole book
- Read page at a time

### Finite fields
- Works well
- Not that I know of
## Comparisons

<table>
<thead>
<tr>
<th>Method</th>
<th>Comparison table</th>
<th>Numerical</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Symbolic</td>
<td></td>
</tr>
<tr>
<td>Exact</td>
<td>Yes</td>
<td>Not really (but certifiable)</td>
</tr>
<tr>
<td># variables</td>
<td>matters a lot</td>
<td>doesn’t matter much</td>
</tr>
<tr>
<td>degrees of polynomials</td>
<td>matters a bit</td>
<td>matters</td>
</tr>
</tbody>
</table>
## Comparisons

<table>
<thead>
<tr>
<th>Method</th>
<th>Comparison table</th>
<th>Numerical</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Symbolic</td>
<td></td>
</tr>
<tr>
<td>Exact</td>
<td>Yes</td>
<td>Not really (but certifiable)</td>
</tr>
<tr>
<td># variables</td>
<td>matters a lot</td>
<td>doesn’t matter much matters</td>
</tr>
<tr>
<td>degrees of polynomials</td>
<td>matters a bit</td>
<td></td>
</tr>
<tr>
<td>Parallelizable</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Works well over C

Easy to use

Largest system I’ve solved ≈ 5000
Can certify real roots Yes
Can certify rational roots Yes

Works well

Finite fields

Not that I know of

Download whole book

Read page at a time

Metaphor
## Comparisons

<table>
<thead>
<tr>
<th>Method</th>
<th>Comparison table</th>
<th>Numerical</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Symbolic</td>
<td></td>
</tr>
<tr>
<td>Exact</td>
<td>Yes</td>
<td>Not really (but certifiable)</td>
</tr>
<tr>
<td># variables</td>
<td>matters a lot</td>
<td>doesn’t matter much</td>
</tr>
<tr>
<td>degrees of polynomials</td>
<td>matters a bit</td>
<td>matters</td>
</tr>
<tr>
<td>Parallelizable</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Works well over $\mathbb{C}$</td>
<td>Not really...</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Comparisons

<table>
<thead>
<tr>
<th>Method</th>
<th>Comparison table</th>
<th>Numerical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact</td>
<td></td>
<td>Not really (but certifiable)</td>
</tr>
<tr>
<td># variables</td>
<td>Yes</td>
<td>doesn’t matter much matters</td>
</tr>
<tr>
<td>degrees of polynomials</td>
<td>matters a lot</td>
<td>Yes</td>
</tr>
<tr>
<td>Parallelizable</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Works well over $\mathbb{C}$</td>
<td>Not really...</td>
<td>Yes</td>
</tr>
<tr>
<td>Easy to use</td>
<td>Definitely</td>
<td>Takes some engineering</td>
</tr>
</tbody>
</table>

- Worked largest system: $\approx 5000$ for symbolic methods and $\approx 200000$ for numerical methods.
- Can certify real roots: Yes for both symbolic and numerical methods.
- Can certify rational roots: Yes for symbolic methods, working on it for numerical methods.
- Metaphor: Download whole book vs. read page at a time.
- Works well over finite fields: Works well for symbolic methods, not that I know of for numerical methods.
## Comparisons

<table>
<thead>
<tr>
<th>Method</th>
<th>Comparison table</th>
<th>Numerical</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Symbolic</td>
<td></td>
</tr>
<tr>
<td>Exact</td>
<td>Yes</td>
<td>Not really (but certifiable)</td>
</tr>
<tr>
<td># variables</td>
<td>matters a lot</td>
<td>doesn’t matter much</td>
</tr>
<tr>
<td>degrees of polynomials</td>
<td>matters a bit</td>
<td>matters</td>
</tr>
<tr>
<td>Parallelizable</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Works well over ( \mathbb{C} )</td>
<td>Not really...</td>
<td>Yes</td>
</tr>
<tr>
<td>Easy to use</td>
<td>Definitely</td>
<td>Takes some engineering</td>
</tr>
<tr>
<td>Largest system I’ve solved</td>
<td>( \approx 5000 )</td>
<td>( \approx 200000 )</td>
</tr>
</tbody>
</table>
## Comparisons

<table>
<thead>
<tr>
<th>Method</th>
<th>Comparison table</th>
<th>Numerical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact</td>
<td></td>
<td>Not really (but certifiable)</td>
</tr>
<tr>
<td># variables</td>
<td>Yes</td>
<td>doesn’t matter much</td>
</tr>
<tr>
<td>degrees of polynomials</td>
<td></td>
<td>matters</td>
</tr>
<tr>
<td>Parallelizable</td>
<td></td>
<td>doesn’t matter much</td>
</tr>
<tr>
<td>Works well over $\mathbb{C}$</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Easy to use</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Largest system I’ve solved</td>
<td></td>
<td>Takes some engineering</td>
</tr>
<tr>
<td>Can certify real roots</td>
<td></td>
<td>$\approx 200000$</td>
</tr>
<tr>
<td>Can certify rational roots</td>
<td></td>
<td>Yes</td>
</tr>
</tbody>
</table>

### Symbolic
- Yes
- matters a lot
- matters a bit
- No
- Not really...
- Definitely
- $\approx 5000$
- Yes

### Numerical
- No
- Doesn’t matter much
- Yes
- Yes
- Takes some engineering
- $\approx 200000$
- Yes
## Comparisons

<table>
<thead>
<tr>
<th>Method</th>
<th>Comparison table</th>
<th>Numerical</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Symbolic</td>
<td></td>
</tr>
<tr>
<td>Exact</td>
<td>Yes</td>
<td>Not really (but certifiable)</td>
</tr>
<tr>
<td># variables</td>
<td>matters a lot</td>
<td>doesn’t matter much</td>
</tr>
<tr>
<td>degrees of polynomials</td>
<td>matters a bit</td>
<td>matters</td>
</tr>
<tr>
<td>Parallelizable</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Works well over $\mathbb{C}$</td>
<td>Not really...</td>
<td>Yes</td>
</tr>
<tr>
<td>Easy to use</td>
<td>Definitely</td>
<td>Takes some engineering</td>
</tr>
<tr>
<td>Largest system I’ve solved</td>
<td>$\approx 5000$</td>
<td>$\approx 200000$</td>
</tr>
<tr>
<td>Can certify real roots</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Can certify rational roots</td>
<td>Yes</td>
<td>Working on it</td>
</tr>
</tbody>
</table>
## Comparisons

<table>
<thead>
<tr>
<th>Method</th>
<th>Symbolic</th>
<th>Numerical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact</td>
<td>Yes</td>
<td>Not really (but certifiable)</td>
</tr>
<tr>
<td># variables</td>
<td>matters a lot</td>
<td>doesn’t matter much</td>
</tr>
<tr>
<td>degrees of polynomials</td>
<td>matters a bit</td>
<td>matters</td>
</tr>
<tr>
<td>Parallelizable</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Works well over $\mathbb{C}$</td>
<td>Not really...</td>
<td>Yes</td>
</tr>
<tr>
<td>Easy to use</td>
<td>Definitely</td>
<td>Yes</td>
</tr>
<tr>
<td>Largest system I’ve solved</td>
<td>$\approx 5000$</td>
<td>Takes some engineering</td>
</tr>
<tr>
<td>Can certify real roots</td>
<td>Yes</td>
<td>$\approx 200000$</td>
</tr>
<tr>
<td>Can certify rational roots</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Metaphor</td>
<td>Download whole book</td>
<td>Working on it</td>
</tr>
</tbody>
</table>

Download whole book
Read page at a time
## Comparisons

<table>
<thead>
<tr>
<th>Method</th>
<th>Comparison table</th>
<th>Numerical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact</td>
<td>Yes</td>
<td>Not really (but certifiable)</td>
</tr>
<tr>
<td># variables</td>
<td>matters a lot</td>
<td>doesn’t matter much</td>
</tr>
<tr>
<td>degrees of polynomials</td>
<td>matters a bit</td>
<td>matters</td>
</tr>
<tr>
<td>Parallelizable</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Works well over $\mathbb{C}$</td>
<td>Not really…</td>
<td>Yes</td>
</tr>
<tr>
<td>Easy to use</td>
<td>Definitely</td>
<td>Yes</td>
</tr>
<tr>
<td>Largest system I’ve solved</td>
<td>$\approx 5000$</td>
<td>Takes some engineering</td>
</tr>
<tr>
<td>Can certify real roots</td>
<td>Yes</td>
<td>$\approx 200000$</td>
</tr>
<tr>
<td>Can certify rational roots</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Metaphor</td>
<td>Download whole book</td>
<td>Working on it</td>
</tr>
<tr>
<td>Finite fields</td>
<td>Works well</td>
<td>Not that I know of</td>
</tr>
</tbody>
</table>

- **Symbolic**: Yes
- **Exact**: Yes
- **Numerical**: Not really (but certifiable)
- **Symbolic**: matters a lot
- **Numerical**: doesn’t matter much
- **Symbolic**: matters a bit
- **Numerical**: matters
- **Symbolic**: No
- **Numerical**: Yes
- **Symbolic**: Not really…
- **Numerical**: Yes
- **Symbolic**: Definitely
- **Numerical**: Takes some engineering
- **Symbolic**: $\approx 5000$
- **Numerical**: $\approx 200000$
- **Symbolic**: Yes
- **Numerical**: Yes
- **Symbolic**: Download whole book
- **Numerical**: Working on it
- **Symbolic**: Works well
- **Numerical**: Not that I know of


J. Hauenstein, S. Sherman, and C. Wampler, *Exceptional stewart-gough platforms, segre embeddings, and the special euclidean group*.


W. Stein, *Sage: A computer system for algebra and geometry experimentation*.

Jan Verschelde, *Phcpack: a general-purpose solver for polynomial systems by homotopy continuation*. 